UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/032,667	10/24/2001	Dave Hylands	13270US01	7248

23446 7590 03/07/2006

MCANDREWS HELD & MALLOY, LTD
500 WEST MADISON STREET
SUITE 3400
CHICAGO, IL 60661

EXAMINER	
HO, ANDY	
ART UNIT	PAPER NUMBER
2194	

DATE MAILED: 03/07/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/032,667	**Applicant(s)** HYLANDS ET AL.	
	Examiner Andy Ho	**Art Unit** 2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 December 2005*.

2a)☒ This action is **FINAL**. 2b)□ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-28* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-28* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER

Attachment(s)

1) □ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *9/15/2005*.

4) □ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. This action is in response to the amendment filed 12/13/2005.

2. Claims 1-28 have been examined and are pending in the application.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Admitted Prior Arts (APA) in view of Nelson U.S Patent No. 6,209,061.

As to claim 1, APA teaches a method for generating program overlays from a sequence of program code (overlays of a program, paragraph 12 page 3), each overlay having a set of code and related data contained therein (code/data segments, paragraph 14 page 4), the overlays being transferred from a storage area to a receiving area for processing (overlays being transfers to the memory 120, Fig. 4, paragraph 15 page 4), the method comprising the steps of:

breaking the sequence of program code into a set of segments each contains a certain amount of related code for processing (overlays of a program wherein each overlay segment contains both code/data, paragraph 12 page 3, paragraph 14 page 4);

running a code segment in the set through a linker device (linker technology to create code and data segments, paragraph 16 page 4);

extracting the code segment and related data segment produced by the linker device (code overplays 102-106 and data overplays 108-112, Fig. 1), with each associated pair of code and data segments representing an overlay (code/data overplays A, B and C, Fig. 1).

APA does not explicitly teach an overplay manager, multiple passes of the linker on the segments of code and referencing the overplays.

Nelson teaches a system of using overplay when executing a program (...program 26, during execution, accesses a memory address contained within one of the memory regions within overlay memory 23..., lines 20-22 column 3) wherein the overlays being transferred via an overlay manager from a storage area to a receiving area for processing (...when program 26, during execution, accesses a memory address contained within one of the memory regions within overlay memory 23, overlay memory controller 24 responds to that address..., lines 20-30 column 3; lines 11-52 column 4); linking multiple segments of code (overplay memory 23 can be segmented into multiple segments that are linked together by pointers, lines 9-19 column 3); concatenating the associated code and data segments into paired sets which can be referenced by the overlay manager (...overlay memory 23 can be segmented into a number of memory regions, with each region defined by a base pointer value that evidences a base address of a region of addresses and a length value. A record of each base pointer value is maintained in overlay controller 24. In essence, each base pointer value and an associated length value define a set of storage locations having addresses that are present in main memory 16 but are, for the time being, unused..., lines 10-19 column 3). It would have been obvious to apply the teachings of Nelson to the system of APA because this improves memory performance by providing an overlay memory as disclosed by Nelson (lines 29-47 column 2).

As to claim 2, APA as modified further teaches dividing the code into a common code area, and an overlay code area (code overplays 102-106 and data overplays 108-112, Fig. 1).

As to claim 3, APA as modified further teaches sizing the program code segments so that they will fit within the receiving area (overplays allow program to fit into small memory, paragraph 12 page 3).

As to claim 4, APA as modified further teaches creating stubs being stored in the receiving area for referencing each function in each program code segment (stub in the common area that referencing code and data, paragraph 13 page 3; wrapper functions for a particular overplay, paragraph 15 page 4).

As to claim 5, APA as modified further teaches generating an overlay table to be used in facilitating transfer of the overlays from the storage area to the receiving area, the overlay table being stored in the receiving area (code overplay area 126 and data overplay area 128 of the memory area 120, Fig. 1).

As to claim 6, APA as modified further teaches the storage area includes an external storage means (low-MIPS processing unit, paragraph 11 page 3).

As to claim 7, APA as modified further teaches the storage area includes memory associated with a low-MIPS processing device (low-MIPS processing unit, paragraph 11 page 3).

As to claim 8, APA as modified further teaches the receiving area includes memory associated with a high-MIPS processing device (DSP, paragraph 12 page 3).

As to claim 9, APA as modified further teaches the high-MIPS processing device includes a digital signal processor (DSP, paragraph 12 page 3).

As to claim 10, APA as modified further teaches the information is converted into a form usable by a processor (the use of higher level language, paragraph 10 page 3).

As to claim 11, APA as modified further teaches the form includes a source file of a high-level programming language (the use of higher level language, paragraph 10 page 3).

As to claim 12, APA teaches a method for generating program overlays from a sequence of program code (overlays of a program, paragraph 12 page 3), the program code having common code and code to be overlaid and each overlay having a set of code and related data contained therein (code/data segments, paragraph 14 page 4), the overlays being transferred from a storage area to a receiving area for processing (overlays being transfers to the memory 120, Fig. 4, paragraph 15 page 4), the method comprising the steps of:

reserving a memory segment in the receiving area to hold overlaid code and data (code overplays 102-106 and data overplays 108-112, Fig. 1);

breaking the sequence of code to be overlaid into a set of segments, wherein each segment contains a certain amount of related code for processing (overlays of a program wherein each overlay segment contains both code/data, paragraph 12 page 3, paragraph 14 page 4), and each segment is sized to fit in the reserved memory segment (overplays allow program to fit into small memory, paragraph 12 page 3);

creating stubs for each code segment, whereby the stubs represent entry points for functions within each code segment (stub in the common area that referencing code and data, paragraph 13 page 3; wrapper functions for a particular overplay, paragraph 15 page 4);

linking the common code along with the stubs for each code segment (link the wrapper routine with the overlay so when the wrapper is called, it causes the overlay to be transferred, paragraph 13 page 3);

importing symbols from the common code and linking the next individual code segment in the set of segments to produce an image (using linker technology to produce image of linked code and data, paragraph 16 page 4);

extracting overlay code and data from the image produced (code and data being loaded into memory area, paragraph 16 page 4);

APA does not explicitly teach an overplay manager, multiple passes of the linker on the segments of code and referencing the overplays.

Nelson teaches a system of using overplay when executing a program (...program 26, during execution, accesses a memory address contained within one of the memory regions within overlay memory 23..., lines 20-22 column 3) wherein the overlays being transferred via an overlay manager from a storage area to a receiving area for processing (...when program 26, during execution, accesses a memory address contained within one of the memory regions within overlay memory 23, overlay memory controller 24 responds to that address..., lines 20-30 column 3; lines 11-52 column 4); linking multiple segments of code (overplay memory 23 can be segmented into multiple segments that are linked together by pointers, lines 9-19 column 3); concatenating the overlays into a file which can be referenced by the overlay manager (...overlay memory 23 can be segmented into a number of memory regions, with each region defined by a base pointer value that evidences a base address of a region of addresses and a length value. A record of each base pointer value is maintained in overlay controller 24. In essence, each base pointer value and an associated length value define a set of storage locations having addresses that are present in main memory 16 but are, for the time being, unused..., lines 10-19 column 3). It would have been obvious to apply the teachings of Nelson to the system of APA because this improves memory performance by providing an overlay memory as disclosed by Nelson (lines 29-47 column 2).

As to claims 13-18, they are method claims of claims 6-11, respectively. Therefore, they are rejected for the same reasons as claims 6-11 above.

As to claim 19, APA teaches a method for generating program overlays from a sequence of program code (overlays of a program, paragraph 12 page 3), the program code having common code area and overlay code area, each overlay having a set of code and related data contained therein (code/data segments, paragraph 14 page 4), the overlays being transferred from a storage area to a receiving area for processing (overlays being transfers to the memory 120, Fig. 4, paragraph 15 page 4), the method comprising the steps of:

generating a wrapper file (stub in the common area that referencing code and data, paragraph 13 page 3; wrapper functions for a particular overplay, paragraph 15 page 4);

creating a linker command file for the common area and creating a linker command file for the overlay area (link the wrapper routine with the overlay so when the wrapper is called, it causes the overlay to be transferred, paragraph 13 page 3);

performing an initialization for the overlay (calling a function of the overlay, paragraph 15 page 4);

creating a common image for the code and data (using linker technology to produce image of linked code and data, paragraph 16 page 4);

producing overlay sections from the image (areas 126 and 128, Fig. 1);

producing an overlay sections file (file created after linking code and data, paragraph 16 page 4); and

producing a load command file, whereby the command file will load the overlay sections file into the appropriate receiving area (code and data being loaded into memory area, paragraph 16 page 4). APA does not explicitly teach an overplay manager, analyzing the overlay code area and overlay control file.

Nelson teaches a system of using overplay when executing a program (...program 26, during execution, accesses a memory address contained within one of the memory regions within overlay memory 23..., lines 20-22 column 3) wherein the overlays being transferred via an overlay manager from a storage area to a receiving area for processing (...when program 26, during execution, accesses a memory address contained within one of the memory regions within overlay memory 23, overlay memory controller 24 responds to that address..., lines 20-30 column 3; lines 11-52 column 4); concatenating the overlays into a file which can be referenced by the overlay manager (...overlay memory 23 can be segmented into a number of memory regions, with each region defined by a base pointer value that evidences a base address of a region of addresses and a length value. A record of each base pointer value is maintained in overlay controller 24. In essence, each base pointer value and an associated length value define a set of storage locations having addresses that are present in main memory 16 but are, for the time being, unused..., lines 10-19 column 3); analyzing the overlay code area and determining the function entry points for each overlay (the use of pointer for each overlay, lines 2-10 column 4); creating an overlay control file for each overlay describing each pair of code and data associated with each overlay (...by inserting the respective base pointers of the subregions, and the associated length values which define, respectively, the subregion extents, into overlay memory controller 24, addresses issued by CPU 12 onto bus 14 which fall within a subregion, are immediately recognized by overlay memory controller 24..., lines 5-10 column 4). It would have been obvious to apply the teachings of Nelson to the system of APA because this improves memory performance by providing an overlay memory as disclosed by Nelson (lines 29-47 column 2).

As to claim 20, APA as modified further teaches creating a copy of the common image whereby the entry point symbols are removed from the particular overlay to be built and linking together an image for a particular overlay to form an overlay image file (the combination of A-C overlays, Fig. 1); extracting the code and data sections from the overlay image file (separation of code and data overlays in each of A-C, Fig. 1).

As to claim 21, APA as modified further teaches generating a wrapper file reads the overlay control file and generates wrapper functions for each function described therein (stub in the common area that referencing code and data, paragraph 13 page 3; wrapper functions for a particular overplay, paragraph 15 page 4).

As to claim 22, APA as modified further teaches the fault function causes the overlay code and data sections to be paged from the storage area to the receiving area (overlays being transfers to the memory 120, Fig. 4, paragraph 15 page 4). Nelson further teaches an overlay descriptor resides in common data and contains information about the overlay (overlay controller, line 13 column 3); the wrapper function is the entry point to the overlay function (the use of base pointer value, lines 12-19 column 3). Note the discussion of claim 1 above for the reasons of combining the references.

As to claims 23-28, they are method claims of claims 6-11, respectively. Therefore, they are rejected for the same reasons as claims 6-11 above.

Response to Arguments

4. Applicant's arguments filed 12/13/2005 have been fully considered but they are not persuasive.

Applicant argued that Nelson reference does not teach the process of generating overlays (Remarks, last paragraph page 7). In response, APA was used to teach this limitation, not Nelson reference.

Applicant argued that Nelson reference does not teach overlay comprises code (Remarks, last paragraph page 7). In response, APA was used to teach this limitation, not Nelson reference.

Applicant argued that Nelson reference does not teach a linker (Remarks, last paragraph page 7 continue to page 8). In response, APA was used to teach this limitation, not Nelson reference.

Applicant argued that APA does not teach importing symbols from the common code and linking the next individual code segment in the set of segments to produce an image (Remarks, last paragraph page 7 continue to page 8). In response, APA teaches using linker technology to produce image of linked code and data (paragraph 16 page 4). More specifically, APA teaches prior linker technology such that the linker takes object code and libraries to produce the final image (lines 1-3 paragraph 43 page 9). The reference meets the limitation as claimed.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within

TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Andy Ho whose telephone number is (571) 272-3762. A voice mail service is also available for this number. The examiner can normally be reached on Monday – Friday, 8:30 am – 5:00 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Thomson can be reached on (571) 272-3718.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIM) system. Status information for published applications may be obtained from either Private PAIR or' Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 571-272-2100.

Any response to this action should be mailed to:

Commissioner for Patents

P.O Box 1450

Alexandria, VA 22313-1450

Or fax to:

- AFTER-FINAL faxes must be signed and sent to (571) 273 - 8300.
- OFFICAL faxes must be signed and sent to (571) 273 - 8300.
- NON OFFICAL faxes should not be signed, please send to (571) 273 – 3762

A.H
March 5, 2006

WILLIAM THOMSON
SUPERVISORY PATENT EXAMINER